

Rancang Bangun Aplikasi CallTenant dengan Penyimpanan Basis Data untuk Form Dinamis Menggunakan *Framework* Laravel

Rahmatin Nadia, R. V. Hari Ginardi, dan Abdul Munif

Departemen Informatika, Fakultas Teknologi Informasi dan Komunikasi, Institut Teknologi Sepuluh Nopember (ITS)

e-mail: munif.if@gmail.com³

Abstrak— Aplikasi CallTenant adalah aplikasi berbasis web yang digunakan sebagai media penghubung bagi penyelenggara acara (Event) dan pemilik usaha (Tenant). Pada aplikasi CallTenant pihak Event dapat mempublikasikan Event-nya, dan juga membuat formulir pendaftaran sesuai kebutuhannya (form dinamis). Selain itu pihak Event juga dapat melihat laporan Tenant yang mendaftar pada Eventnya. Dengan adanya aplikasi CallTenant diharapkan dapat meningkatkan potensi bisnis atau kerja sama antara penyelenggara acara (Event) dengan pemilik usaha (Tenant). Salah satu fitur yang terdapat pada aplikasi CallTenant adalah pembuatan form dinamis, dimana pihak Event dapat membuat formulir pendaftaran dengan dinamis dalam satu aplikasi tanpa membutuhkan aplikasi tambahan. Bagi pihak Tenant CallTenant menyediakan platform yang dapat mempermudah mereka dalam proses pendaftaran Event, termasuk di dalamnya proses pencarian, pengisian formulir dan juga pembayaran stand. Dalam proses pembuatan form dinamis dibutuhkan rancangan basis data yang sesuai untuk memenuhi kebutuhan bisnis dari CallTenant.

Kata Kunci— Basis Data, CallTenant, Form dinamis, Laravel.

I. PENDAHULUAN

BANYAKNYA *e-commerce* yang bermunculan saat ini memberikan ide untuk membuat salah satu *e-commerce* yang dibutuhkan oleh masyarakat. Terutama mereka-mereka yang menyelenggarakan sebuah acara dan tidak jarang dari mereka yang membutuhkan *Tenant* atau pedagang untuk mengisi di acara mereka. Sejauh ini belum ada *platform* yang dapat menjadi jembatan bagi si pemilik *Event* dan si pemilik *Tenant* untuk saling berbagi informasi. Di sinilah kami memiliki ide untuk membangun aplikasi web *e-commerce* CallTenant.

Selain itu, karena didasarkan pada beragamnya kebutuhan dari masing-masing *Event*, dalam mencari *Tenant* yang sesuai untuk mereka maka diperlukan basis data khusus yang dapat menampung *form* dinamis untuk *Event* tersebut. Seperti yang kita tahu, banyak sekali jenis *Event* yang diselenggarakan. Dalam jurnal ini akan dirancang basis data yang ideal untuk menampung data-data yang dibutuhkan oleh pihak *Event* dari pihak calon *Tenant* yang akan mengisi acara mereka. Diharapkan basis data ini dapat menjadi salah satu sarana penyimpanan data bagi aplikasi CallTenant ke depannya.

Berdasarkan hasil penelusuran pada situs surabaya.go.id terbukti terdapat kurang lebih 90 acara diadakan di kota Surabaya pada tahun 2017, baik acara yang dibuat oleh sebuah promotor besar, oleh mahasiswa, bahkan pelajar sekalipun.

Kesulitan dalam proses pencarian *Tenant* yang akan mengisi acara dan begitu juga sebaliknya, sulitnya *Tenant* mencari *Event* apa yang cocok untuk diikuti oleh mereka. Karena itu aplikasi CallTenant akan berusaha meningkatkan potensi bisnis atau kerja sama diantara kedua belah pihak baik *Event* maupun *Tenant*. Selama ini seperti yang kita tahu apabila sebuah *Event* mencari *Tenant* untuk mengisi stand pada acara mereka, yang mereka lakukan biasanya menempelkan poster atau menyebar poster di media social atau lain sebagainya. Dengan adanya aplikasi CallTenant akan memudahkan kedua belah pihak untuk saling mencari informasi.

II. TINJAUAN PUSTAKA

A. MySQL

MySQL merupakan suatu *software* sistem manajemen basis data yang *open source*. MySQL adalah *database server* yang dibuat dan distribusikan oleh perusahaan komersial yaitu MySQL AB. MySQL didistribusikan secara gratis di bawah lisensi *General Public License* (GPL). MySQL adalah sistem yang mendukung *relational database*. Artinya, dalam sebuah basis data memiliki beberapa tabel untuk menyimpan data—data dimana masing—masing tabel memiliki hubungan atau relasi satu sama lain sehingga dapat dilakukan kombinasi data dari beberapa tabel dalam satu saat. Sistem semacam ini sering disebut pula dengan RDBMS (*Relational DataBase Management Sistem*). Sistem manajemen basis data seperti MySQL diperlukan untuk menambahkan, mengakses, memproses data yang disimpan di server [1].

Secara umum ada 3 jenis query pada mysql yaitu DDL (*Data Definition Language*), DML (*Data Manipulation Language*) dan DCL (*Data Control Language*). Pada pembangunan aplikasi CallTenant, MYSQL digunakan bersamaan dengan skrip PHP untuk membuat aplikasi yang dinamis dan powerful.

B. Dynamic Form

Dynamic form field adalah jumlah *field* yang dinamis dalam form. Yang di maksud dinamis di sini adalah form yang akan di buat kali ini adalah sebuah form input yang bisa bertambah atau pun di kurangi jumlah inputannya sesuai dengan kebutuhan *User*. Salah satu contoh aplikasi pembuatan form dinamis yang umum digunakan adalah Google Form. Terdapat *library* yang dapat membantu pengguna untuk membangun aplikasi yang memiliki form dinamis, seperti AngularJS. Selain itu, terdapat juga *platform* yang

menyediakan template form yang dapat langsung digunakan oleh pengguna diantaranya TCEXAM dan Google Form.

1) *Angular*

AngularJS adalah sebuah *framework* Javascript yang di kembangkan atau di bangun oleh tim Google. AngularJS menggunakan metode MVC yang membuat source code aplikasi kita menjadi bersih dan mudah di kembangkan. sampai saat ini AngularJS sudah menjadi salah satu *framework* Javascript yang paling populer dan sangat banyak di gunakan oleh para developer di seluruh dunia untuk membangun aplikasi .

NG Dynamic Forms adalah *library* pengembangan form yang cepat berdasarkan panduan Angular *dynamic form*. NG Dynamic Forms sepenuhnya mengotomatisasi pembentukan form UI dengan memperkenalkan seperangkat model *form control* yang dapat dipelihara dan komponen *dynamic form control*. *Library* ini mendukung beberapa *library* UI populer seperti Bootstrap, Foundation, Ionic, Kendo, Material, NG Bootstrap, ngx-bootstrap dan PrimeNG [12].

2) *Google Form*

Google Forms - bersama dengan Documents, Spreadsheet, dan Slide - adalah bagian dari aplikasi online Google untuk membantu Anda menyelesaikan banyak hal di browser Anda secara gratis. Mudah digunakan dan salah satu cara termudah untuk menyimpan data langsung ke spreadsheet [4] [2].

Google Form atau yang disebut google formulir adalah alat yang berguna untuk membantu user dalam membuat sebuah form dengan dinamis yang dapat digunakan untuk merencanakan acara, mengirim survei, memberikan kuis, atau mengumpulkan informasi yang mudah dengan cara yang efisien [6]. Google Forms adalah salah satu layanan termudah untuk digunakan. Ini mendukung banyak bahasa dan secara otomatis membuat visualisasi [3].

3) *TCEXAM*

TCEXAM adalah salah satu aplikasi Computer-based Assesment gratis yang dapat dimanfaatkan untuk melaksanakan ujian berbasis komputer dan jaringan atau bahkan internet (online). TCEXAM dibangun dengan bahasa pemrograman PHP dan mendukung database MySQL. TCEXAM mudah dikembangkan dan digunakan untuk ujian online [1].

Perbandingan dari beberapa sumber diatas mengenai form dinamis dapat dilihat pada Tabel 1. Dapat dilihat bahwa Google Form memiliki lebih banyak keunggulan dibanding dengan aplikasi sejenis yang lain, tetapi modul form dinamis pada CallTenant menyesuaikan dengan kebutuhan aplikasi yang akan dibuat, dalam hal ini User membuat form dinamis

| | | | | |
|-----------------------|---|---|---|---|
| <i>Dropdown</i> | ✗ | ✓ | ✓ | ✓ |
| <i>Short Text</i> | ✓ | ✓ | ✓ | ✓ |
| <i>Long Text</i> | ✓ | ✓ | ✓ | ✓ |
| <i>Linear Scale</i> | ✗ | ✗ | ✓ | ✗ |
| <i>Ordering</i> | ✓ | ✗ | ✓ | ✗ |
| <i>File Upload</i> | ✗ | ✗ | ✓ | ✗ |
| <i>Multi Page</i> | ✗ | ✗ | ✓ | ✗ |
| <i>Editor</i> | ✓ | ✗ | ✓ | ✓ |
| <i>Delete Field</i> | ✓ | ✓ | ✓ | ✓ |
| <i>Move up/down</i> | ✗ | ✓ | ✓ | ✓ |
| <i>Add new option</i> | ✓ | ✓ | ✓ | ✓ |
| <i>Delete option</i> | ✓ | ✓ | ✓ | ✓ |

yang hanya akan digunakan untuk registrasi pendaftaran Tenant.

III. DESAIN SISTEM

Pada aplikasi terdapat fitur pembuatan formulir pendaftaran secara dinamis. Bab ini akan menjelaskan cara kerja dari fitur tersebut.

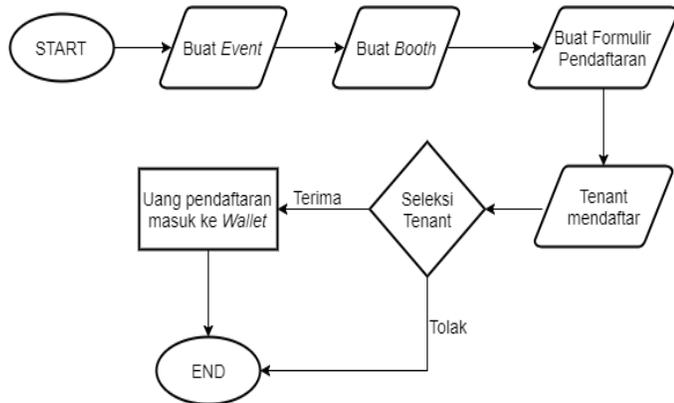
A. *Deskripsi Umum Aplikasi*

CallTenant adalah aplikasi berbasis web yang digunakan sebagai media penghubung bagi penyelenggara acara (*Event*) dan pemilik usaha (*Tenant*). Pada aplikasi CallTenant pihak *Event* dapat mempublikasikan *Event*-nya, dan juga membuat formulir pendaftaran sesuai kebutuhannya (form dinamis). Selain itu pihak *Event* juga dapat melihat laporan *Tenant* yang mendaftar pada *Event*-nya. Dalam melihat laporan, *Event* dapat memilih apakah hendak melihat rekap pendaftar secara keseluruhan ataupun dikelompokkan berdasar pertanyaan yang diajukan ataupun berdasarkan *Tenant* yang mendaftar. Dari sisi pemilik usaha, *Tenant* dapat mencari *Event* sesuai kategori ataupun sesuai namanya.

Setelah melakukan pencarian *Tenant* juga dapat mendaftar pada *Event* yang diinginkan dengan cara mengisi form pendaftaran, sebelum proses pengisian formulir pendaftaran *Tenant* dapat memilih jenis *Booth* yang tersedia. Aplikasi CallTenant juga menyediakan fitur *Review* baik untuk *Event* ataupun *Tenant* , FAQ dan *Contact us*, yang diharapkan dapat membantu *User* apabila memiliki pertanyaan terkait dengan aplikasi CallTenant. *User* yang telah terdaftar pada aplikasi CallTenant dapat memiliki lebih dari satu *Event* atau lebih dari satu *Tenant*, tergantung pada *role*-nya, tetapi satu *User* hanya dapat menjadi *Event* atau *Tenant* saja.

Tabel 1. Perbandingan Aplikasi Form Dinamis

| Kategori | TCEXAM | Angular (NG Dynamic Forms) | Google Form | CallTenant |
|---------------------|--------|----------------------------|-------------|------------|
| <i>Checkbox</i> | ✗ | ✓ | ✓ | ✓ |
| <i>Datepicker</i> | ✗ | ✓ | ✓ | ✓ |
| <i>Radio Button</i> | ✓ | ✓ | ✓ | ✓ |



Gambar 1 Diagram Alir User Event

Gambar 1 menjelaskan mengenai alur proses penerimaan pendaftaran *Tenan* bagi *User Event*. Proses pembuatan formulir dimulai dengan membuat *Event* dan *Booth*. Setelah itu *User Event* memilih *Event* mana yang akan dibuat formulir pendaftarannya dan *User* dapat membuat formulir pendaftaran sesuai yang dibutuhkan. Setelah formulir berhasil dibuat maka *Tenan* dapat mendaftarkan diri pada *Event* tersebut. *Event* dapat menerima atau menolak *Tenan* sesuai dengan keinginannya. Setelah *Tenan* diterima maka uang pendaftaran akan masuk ke *wallet Event*, dan baik pihak *Event* maupun *Tenan* akan mendapatkan *point* yang dapat ditukarkan dengan *Reward*.

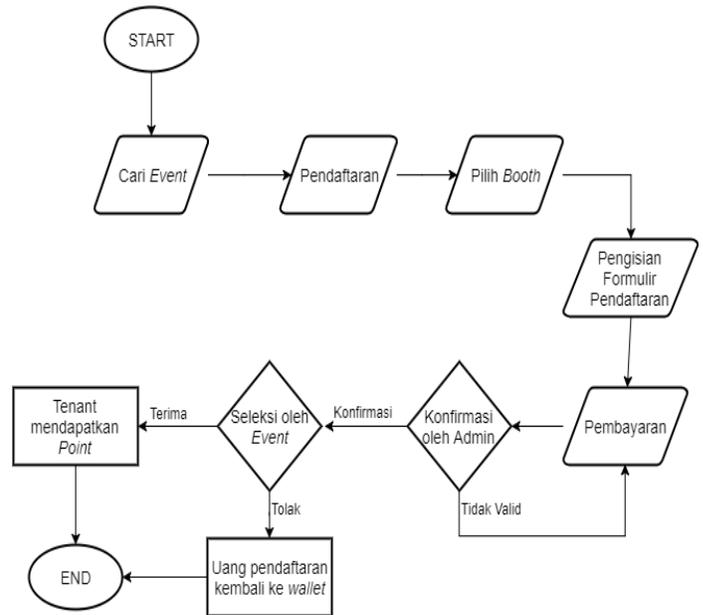
Gambar 2 menjelaskan mengenai proses pendaftaran *Tenan* pada *Event*. Pertama *Tenan* melakukan pencarian *Event* dan memilih *Event* mana yang akan diikuti. Setelah itu *Tenan* dapat memilih *Booth* untuk melakukan pendaftaran pada *Event* berikutnya adalah mengisi formulir pendaftaran dan melakukan pembayaran. Setelah pembayaran selesai dan telah dikonfirmasi oleh Admin maka *Tenan* dinyatakan telah terdaftar pada *Event* tersebut. Pihak *Event* dapat menerima ataupun menolak *Tenan* yang mendaftar, apabila *Tenan* diterima maka *Tenan* akan mendapatkan *point*, sedangkan apabila *Tenan* ditolak oleh *Event* maka uang pendaftaran akan kembali ke *wallet Tenan*.

B. Implementasi Elemen Form Dinamis

Pada aplikasi *CallTenant user Event* dapat membuat form pendaftaran sesuai kebutuhannya, atau dengan kata lain dengan dinamis, selain membuat *User Event* juga dapat mengubah formulir yang telah dibuat sebelumnya. Elemen form dinamis yang dapat dipilih oleh *User* dapat dilihat pada Tabel 2.

Untuk tipe jawaban *Radio Button*, *Check Box*, dan *Select User* dapat menambahkan *option* jawaban sebanyak yang mereka inginkan. Selain menambah *User* juga dapat menghapus *option* jawaban tersebut. Selain itu pada masing-masing pertanyaan terdapat atribut *mandatory* atau pertanyaan yang wajib diisi oleh pihak *Tenan*. *User* dapat menentukan sendiri apakah pertanyaan tersebut wajib diisi atau tidak.

Apabila masing-masing *button* ditekan maka akan muncul panel untuk mengisi pertanyaan dan memilih apakah pertanyaan tersebut wajib diisi atau tidak (*required or not*). Yang membedakan tipe jawaban *Radio Button*, *Check Box*, dan *Select* dengan yang lain adalah terdapat tombol *add option*



Gambar 2 Diagram Alir Proses Pendaftaran Tenan pada Event

untuk menambah pilihan jawaban dan *add other* untuk menambah pilihan jawaban *other* yang dapat diisi sendiri oleh *User*. Detail dari masing-masing fungsionalitas *button* jawaban dapat dilihat pada Tabel 3.

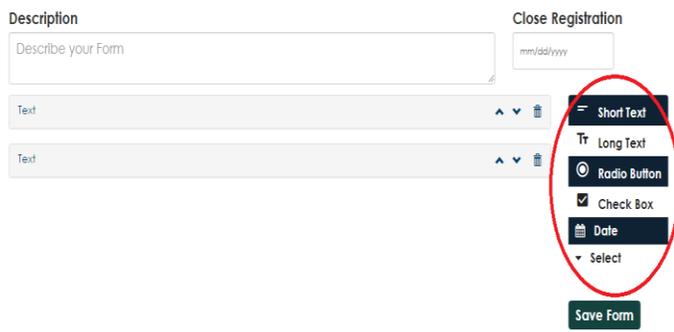
C. Implementasi Basis Data

CallTenant memiliki 16 tabel yang saling berhubungan dan dapat dilihat pada Gambar 4. Tabel *User* digunakan untuk menyimpan data *User CallTenant*, baik dengan *role Event* atau *Tenan*. Tabel *Event* dan *Tenan* berguna untuk menyimpan data *Event* dan *Tenan*, sedangkan untuk menyimpan gambar dari masing-masing *User Event* dan *Tenan* terdapat pada tabel *event_image* dan *tenant_image*. *Event* juga memiliki tabel untuk menyimpan data *Booth* yang dimiliki termasuk harga dan kapasitas dari *Booth*.

Untuk menyimpan data-data form dinamis, *Event* memiliki 3 tabel, yaitu tabel formulir, pertanyaan dan *radio_check*. Tabel formulir berisikan data *id_event* dan deskripsi dari formulir pendaftaran tersebut. Tabel pertanyaan berisikan data pertanyaan dari formulir pendaftaran, termasuk di dalamnya terdapat jenis pertanyaan

Tabel 2. Implementasi Elemen Form Dinamis

| Jenis | Penjelasan |
|---------------------|---|
| Short Text | Pilihan jawaban untuk <i>text</i> pendek dengan panjang jawaban yang dapat ditampung adalah 256 karakter |
| Long Text | Pilihan jawaban untuk <i>text</i> yang relatif panjang dengan panjang jawaban yang dapat ditampung adalah 1024 karakter |
| Radio Button | Tombol yang digunakan untuk mengambil salah satu pilihan dari banyak pilihan yang tersedia |
| Check Box | Tombol yang memungkinkan <i>User</i> untuk dapat memilih daftar lebih dari satu pilihan jawaban |
| Date | Pilihan jawaban berupa tanggal |
| Select | Tombol yang digunakan untuk mengambil salah satu pilihan dari banyak pilihan yang tersedia berupa list ke bawah |



Gambar 3 Tampilan membuat formulir pendaftaran

Tabel 3.
Detail Elemen Form Dinamis

| Fungsionalitas | Jenis | | | | | |
|-------------------------------|------------|-----------|--------------|-----------|------|--------|
| | Short Text | Long Text | Radio Button | Check Box | Date | Select |
| Tambah Pertanyaan | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Hapus Pertanyaan | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Move Up/Down | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Mandatoryfield | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Tambah dan hapus Option | - | - | ✓ | ✓ | - | ✓ |
| Tambah dan hapus Other Option | - | - | ✓ | ✓ | - | - |

dan keterangan apakah pertanyaan tersebut wajib diisi atau tidak dan tanggal pendaftaran dari *Event* tersebut. Berikutnya, tabel *radio_check* berisikan data jawaban dari pertanyaan dengan jenis *radio button*, *check box* dan *select*.

Untuk menyimpan data jawaban *Tenant* terdapat pada tabel jawaban, tabel ini memiliki relasi dengan tabel pertanyaan untuk mengetahui pertanyaan mana yang dijawab oleh *Tenant*. Apabila *Tenant* mendaftar pada *Event* tertentu maka data pendaftaran akan masuk ke tabel *request*, dan untuk pemesanan *Booth* akan masuk ke tabel *request_booth* yang di dalamnya terdapat *id_booth* dan jumlah *booth* yang dipesan. Tabel *request*, *request_booth* dan jawaban. Tabel pembayaran berguna untuk menyimpan data pembayaran dari *Tenant*, dan tabel *payment_confirmation* berisi data konfirmasi pembayaran yang dilakukan oleh *Tenant*.

D. Implementasi Proses Membuat Formulir Pendaftaran

Pada proses ini, *User Event* dapat membuat formulir pendaftaran yang nantinya akan diisi oleh *User Tenant*. Pada Gambar 1 proses pembuatan formulir dimulai dengan membuat *Event* dan *Booth* yang dimiliki oleh *Event* tersebut. Setelah itu *User Event* memilih *Event* mana yang akan dibuat formulir pendaftarannya dan *User* dapat membuat formulir pendaftaran sesuai yang dibutuhkan. Setelah formulir berhasil dibuat maka *Tenant* dapat mendaftarkan diri pada *Event* tersebut. *Event* dapat menerima atau menolak *Tenant* sesuai dengan keinginannya. Setelah *Tenant* diterima maka uang pendaftaran akan masuk ke *wallet Event*, dan baik pihak *Event* maupun *Tenant* akan mendapatkan *point* yang dapat ditukarkan dengan *Reward*.

Gambar 3 merupakan tampilan membuat formulir pendaftaran bagi *Event*. Halaman *Create Form* berisi input data deskripsi dan tanggal penutupan pendaftaran *Event*. Terdapat tombol *Save Form* untuk menyimpan formulir pendaftaran yang telah dibuat. Terdapat juga tombol-tombol tipe jawaban untuk pertanyaan yang diajukan pada formulir registrasi pendaftaran. Pada masing-masing tombol tipe jawaban terdapat *input* pertanyaan dan tombol *Move Up*, *Move Down* yang berguna untuk merubah urutan posisi pertanyaan dan tombol *Delete* untuk menghapus pertanyaan. Sedangkan khusus untuk tipe jawaban *check box*, *radio button* dan *select* terdapat tambahan tombol *Add Option* untuk menambah jawaban yang telah dijelaskan pada Tabel 3.

E. Implementasi Mengubah Formulir Pendaftaran

Halaman *Edit Form* merupakan halaman yang sama dengan *Create Form*, tetapi telah berisi data-data pertanyaan yang telah dibuat sebelumnya. Setelah *User* selesai melakukan perubahan formulir pendaftaran maka *User Event* dapat menekan tombol *Save Form* untuk menyimpan perubahan. *User* dapat menghapus menambah maupun mengubah pertanyaan yang terdapat pada halaman tersebut.

IV. PENGUJIAN DAN EVALUASI

Pengujian dilakukan oleh 20 orang di mana sepuluh orang akan berperan sebagai *Event* dan sepuluh orang akan berperan sebagai *Tenant*. Setiap pengujian dilakukan secara bersamaan oleh 2 orang yang berperan menjadi *Event* dan *Tenant*. Setelah itu, responden mengisi kuesioner untuk menguji ketertarikan terhadap kebutuhan fungsional pada aplikasi CallTenant. Penilaian kuesioner pada setiap pertanyaannya diproyeksikan dalam rating dengan skala 1-5.

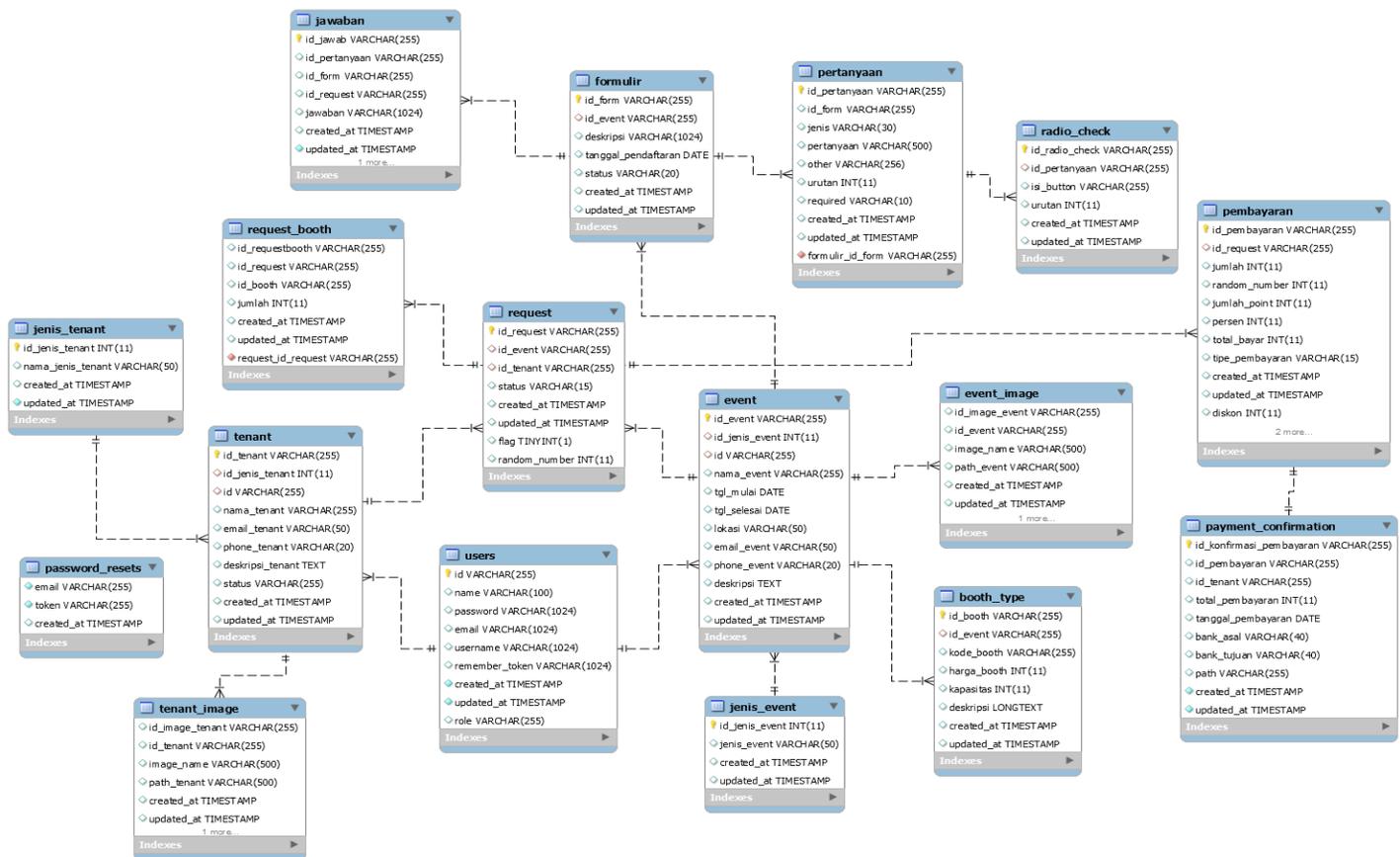
Hasil evaluasi skor pada Tabel 4 dan Tabel 5 didapatkan dari penghitungan *Confidence Interval*. *Confidence Interval* adalah rentang antara dua nilai di mana nilai suatu *Sample Mean* tepat berada di tengah-tengahnya. Nilai sebuah *confidence interval* dapat dinyatakan dengan Kemungkinan (*Probability*) berapa *sample* dalam 100 kali pengambilan *samples* nilai *Population Mean* sesungguhnya akan masuk dalam sebuah rentang *sample mean* [10].

Confidence interval (CI) menggunakan rumus berikut :

$$CI = \bar{x} \pm Z \frac{s}{\sqrt{n}} \tag{1}$$

Persamaan (1) merupakan rumus *confidence interval*, dimana \bar{x} adalah rata-rata dari nilai kuesioner, Z adalah nilai koefisien *confidence*, s adalah standar deviasi, dan n adalah jumlah responden.

Hasil penghitungan pengujian evaluasi dapat dilihat pada Tabel 4 untuk *Event* dan pada Tabel 5 untuk pihak *Tenant*. Status “OK” didapatkan apabila nilai persentase *error* kurang dari 21.9%, maka dianggap responden setuju dengan hasil survei yang telah diberikan dan apabila persentase *error* lebih dari 21.9% maka dianggap responden tidak setuju dengan hasil kuesioner yang telah diberikan. Nilai 21.9 % persen didapatkan dari penghitungan *error level* pada akurasi survei [9].



Gambar 4. Implementasi Basis Data CallTenant

Tabel 4. Hasil Evaluasi Kuesioner Responden Event

| No | Fitur | Rata-rata | Standar Deviasi | Confidence Interval | | | Persentase Error | Status |
|----|--------------------------------|-----------|-----------------|---------------------|--------|--------|------------------|--------|
| | | | | CI | Mean ↑ | Mean ↓ | | |
| 1 | Pembuatan formulir pendaftaran | 4.4 | 0.5164 | 0.3201 | 4.0799 | 4.7201 | 8.00% | OK |
| 2 | Ubah formulir | 4.5 | 0.5270 | 0.3267 | 4.1733 | 4.8267 | 8.17% | OK |
| 3 | Laporan | 4.5 | 0.5270 | 0.3267 | 4.1733 | 4.8267 | 8.17% | OK |
| 4 | Mengelola Booth | 4.4 | 0.8433 | 0.5227 | 3.8773 | 4.9227 | 13.07% | OK |

Tabel 5. Hasil Evaluasi Kuesioner Responden Tenant

| No | Fitur | Rata-rata | Standar Deviasi | Confidence Interval | | | Persentase Error | Status |
|----|------------------------------------|-----------|-----------------|---------------------|--------|--------|------------------|--------|
| | | | | CI | Mean ↑ | Mean ↓ | | |
| 1 | Pencarian Event | 4.1 | 0.568 | 0.352 | 3.748 | 4.452 | 8.80% | OK |
| 2 | Proses pendaftaran Event | 4.5 | 0.707 | 0.438 | 4.062 | 4.938 | 10.96% | OK |
| 3 | Mengisi formulir pendaftaran Event | 4.1 | 0.568 | 0.352 | 3.748 | 4.452 | 8.80% | OK |

V. KESIMPULAN DAN SARAN

Paper ini memperkenalkan aplikasi baru, CallTenant. CallTenant merupakan berbasis web yang digunakan sebagai media penghubung bagi penyelenggara acara (Event) dan pemilik usaha (Tenant). Pihak Event dapat mempublikasikan Event-nya, dan juga membuat formulir pendaftaran sesuai kebutuhannya (form dinamis). Dari pihak Tenant dapat mencari Event sesuai kategori ataupun sesuai namanya. Setelah melakukan pencarian Tenant juga dapat mendaftar pada Event yang diinginkan dengan cara mengisi form pendaftaran dan melakukan pembayaran. Aplikasi CallTenant

mampu mempermudah proses bisnis antara pihak Tenant dan pihak Event, dalam hal ini difokuskan pada proses pendaftaran.

Saran untuk pembangunan aplikasi ke depannya, diantaranya menambahkan fitur rekomendasi yang dapat dilihat dari track record Tenant sebelumnya, dapat juga ditambahkan fitur Reminder atau pengingat bagi Event dan Tenant. Akan lebih baik lagi jika dibuat mobile app CallTenant. Dan menambahkan sekuritas pada proses pembayaran.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih Laboratorium Manajemen Informasi dan Departemen Informatika Institut Teknologi Sepuluh Nopember yang telah mendukung dan memfasilitasi penelitian ini.

DAFTAR PUSTAKA

- [1] Asuni, N. (2017). *TCEXAM*. (Tecnick.com) Dipetik Januari 17, 2018, dari <https://tcexam.org/>
- [2] *Create Forms*. (t.thn.). (Google.com) Dipetik Januari 17, 2018, dari <https://www.google.com/forms/about/>
- [3] *Google Forms*. (2017, Juni 26). (Peace Tech Wiki) Dipetik Desember 31, 2017, dari http://peacetech.wiki/index.php?title=Google_Forms
- [4] Guay, M. (t.thn.). *Google Forms Guide: Everything You Need to Make Great Forms for Free*. (zapier) Dipetik Desember 31, 2017, dari <https://zapier.com/learn/google-sheets/how-to-use-google-forms/>
- [5] Johan, Tulenan, V., & Sambul, A. M. (2016). Rancang Bangun Aplikasi Pendataan Potensi Desa / Kelurahan Berbasis Web. *Teknik Informatika Universitas Sam Ratulangi Manado*, 9.
- [6] *Mengenal Google Form untuk Kebutuhan Survey Anda*. (2016, Agustus 16). (idCloudHost) Dipetik Desember 30, 2017, dari <https://idcloudhost.com/mengenal-google-form-untuk-kebutuhan-survey-anda/>
- [7] *MySQL 5.7 Reference Manual*. (2017, December 29). Dipetik Desember 30, 2017, dari <https://downloads.mysql.com/docs/refman-5.7-en.pdf>
- [8] Sunyoto, A. (2007). *Ajax Membangun Web dengan Teknologi Asynchronous JavaScript&XML*. Yogyakarta: ANDI OFFSET.
- [9] *Survey Random Sample Calculator*. (2018). (costum insight) Dipetik Januari 7, 2018, dari <https://www.custominsight.com/articles/random-sample-calculator.asp>
- [10] *SURVEY STATISTICAL CONFIDENCE: HOW MANY IS ENOUGH?*. (GreatBrook) Dipetik Januari 7, 2018, dari <https://greatbrook.com/survey-statistical-confidence-how-many-is-enough/>
- [11] Tang, L., Li, T., Jiang, Y., & Chen, Z. (2013). Dynamic Query Forms for Database Queries. 26, hal. 2166-2178. *IEEE transactions on knowledge and data engineering*.
- [12] udos86. (2017, Desember). *NG Dynamic Forms*. (Github) Dipetik Desember 31, 2017, dari <https://github.com/udos86/ng-dynamic-forms>
- [13] Welling, L., & Thomson, L. (2009). *PHP and MySQL Web Development 4th Edition*. Pearson Education.